

**METHOD AND SYSTEM FOR DISTRIBUTING REQUESTS FOR CONTENT**

**Field of the Invention**

5 The present invention relates to information caching, and in particular to caching content from servers distributed over a wide area network.

**Background**

The World Wide Web (WWW) provides access to a vast amount of information. At times, some information is accessed much more frequently than other 10 information. At other times, information that was frequently accessed previously becomes less frequently accessed. For example, when a hot news story is first published on the WWW, many users may request the Web page featuring the story. As the story becomes older and other stories are published, the Web page may be requested less frequently. Eventually, the Web page may be requested seldom if at all.

15 Similarly, a company may announce a new product and post information on its Web site about the product. Initially, many users may request Web pages regarding the product. These requests may follow a certain pattern each day. For example, during morning working hours, requests for Web pages regarding the product may spike. Later in the day, requests for the Web pages may decrease. At night, 20 relatively few requests for the pages may be received.

A company may place information on a Web page for a variety of reasons. For example, it may place information on Web pages to attract potential customers, inform employees, or establish a presence on the Web. When a user is required to wait too long for a Web page to be served, however, the user often loses 25 interest and decides to visit other Web sites or pages. Responding too slowly to Web page requests may be harmful for sales, decrease worker productivity, or give a company a bad image.

**Summary**

In accordance with the present invention, there is provided a method and system for caching content, such as content requested from a server on the WWW. Requests for content that may change for each user (also known as dynamic content) are 5 forwarded to a content server to avoid caching data that might only be used once. Requests for static content are forwarded to a hot or a regular cache, depending on the frequency at which the content is requested. When a hot cache does not contain the content, it requests the content from the forwarder which then forwards the request to a regular cache. When a regular cache does not contain the content, it requests the 10 content from the forwarder which then forwards the request to a content server. There may be more than two layers of cache.

In one aspect of the invention, the caches do not need to be aware of what content other caches contain. Rather, when a cache does not have requested content, it requests the content from the forwarder which then forwards it to another 15 cache or to a content server as appropriate.

In another aspect of the invention, content that is frequently requested is forwarded to a hot cache. If the hot cache does not contain the content, it sends a request to the forwarder which forwards the request to a regular cache which would likely have the content. If the regular cache does not have the content, it sends a request 20 to the forwarder which then sends the request to a content server.

These and various other features as well as advantages, which characterize the present invention, will be apparent from a reading of the following detailed description and a review of the associated drawings.

**Brief Description of the Drawings**

25 FIGURES 1-3 show components of an exemplary environment in which the invention may be practiced;

FIGURE 4 illustrates an exemplary environment in which the invention operates in which a virtual server array is connected to a server over a communications medium;

FIGURE 5 shows a caching mechanism for speeding access to content;  
FIGURE 6 illustrates an exemplary table that may be used to maintain  
information about content requests;

FIGURE 7 shows a flow chart for receiving and forwarding requests for  
5 content;

FIGURE 8 illustrates a flow chart for determining where to forward a  
request for non-dynamic content;

FIGURE 9 shows a flow chart for performing a hot/cold calculation; and

FIGURE 10 illustrates an exemplary environment in which the invention  
10 operates in which a virtual server array is connected to a server over a network in  
accordance with the invention.

### Detailed Description

In the following detailed description of exemplary embodiments of the  
invention, reference is made to the accompanied drawings, which form a part hereof,  
15 and which are shown by way of illustration, specific exemplary embodiments of which  
the invention may be practiced. These embodiments are described in sufficient detail to  
enable those skilled in the art to practice the invention, and it is to be understood that  
other embodiments may be utilized, and other changes may be made, without departing  
from the spirit or scope of the present invention. The following detailed description is,  
20 therefore, not to be taken in a limiting sense, and the scope of the present invention is  
defined only by the appended claims.

In the following description, first an illustrative operating environment in  
which the invention may be practiced is disclosed. Then, an exemplary caching system  
operating in the illustrative operating environment is described. Next, a data structure  
25 that may be used to calculate the frequency of requests for content is described. Then a  
method of using the above to forward requests for content is given. Finally, another  
configuration of an exemplary caching system is described.

### Illustrative Operating Environment

FIGURES 1-3 show components of an exemplary environment in which the invention may be practiced. Not all the components may be required to practice the invention, and variations in the arrangement and type of the components may be made without departing from the spirit or scope of the invention.

5 FIGURE 1 shows a plurality of local area networks ("LANs") 120<sub>a-d</sub> and wide area network ("WAN") 130 interconnected by routers 110. Routers 110 are intermediary devices on a communications network that expedite message delivery. On a single network linking many computers through a mesh of possible connections, a router receives transmitted messages and forwards them to their correct destinations  
10 over available routes. On an interconnected set of LANs--including those based on differing architectures and protocols--, a router acts as a link between LANs, enabling messages to be sent from one to another. Communication links within LANs typically include twisted pair, fiber optics, or coaxial cable, while communication links between networks may utilize analog telephone lines, full or fractional dedicated digital lines  
15 including T1, T2, T3, and T4, Integrated Services Digital Networks (ISDNs), Digital Subscriber Lines (DSLs), wireless links, or other communications links known to those skilled in the art. Furthermore, computers, such as remote computer 140, and other related electronic devices can be remotely connected to either LANs 120<sub>a-d</sub> or  
20 WAN 130 via a modem and temporary telephone link. The number of WANs, LANs, and routers in FIGURE 1 may be increased or decreased arbitrarily without departing from the spirit or scope of this invention.

As such, it will be appreciated that the Internet itself may be formed from a vast number of such interconnected networks, computers, and routers. Generally, the term "Internet" refers to the worldwide collection of networks, gateways, routers, and  
25 computers that use the Transmission Control Protocol/Internet Protocol ("TCP/IP") suite of protocols to communicate with one another. At the heart of the Internet is a backbone of high-speed data communication lines between major nodes or host computers, including thousands of commercial, government, educational, and other computer systems, that route data and messages. An embodiment of the invention may

be practiced over the Internet without departing from the spirit or scope of the invention.

The media used to transmit information in communication links as described above illustrates one type of computer-readable media, namely

5 communication media. Generally, computer-readable media includes any media that can be accessed by a computing device. Computer-readable media may include computer storage media, communication media, or any combination thereof.

Communication media typically embodies computer-readable instructions, data structures, program modules, or other data in a modulated data signal

10 such as a carrier wave or other transport mechanism and includes any information delivery media. The term “modulated data signal” means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, communication media includes wired media such as twisted pair, coaxial cable, fiber optics, wave guides, and other wired media and

15 wireless media such as acoustic, RF, infrared, and other wireless media.

The Internet has recently seen explosive growth by virtue of its ability to link computers located throughout the world. As the Internet has grown, so has the WWW. Generally, the WWW is the total set of interlinked hypertext documents residing on HTTP (hypertext transport protocol) servers around the world. Documents

20 on the WWW, called pages or Web pages, are typically written in HTML (Hypertext Markup Language) or some other markup language, identified by URLs (Uniform Resource Locators) that specify the particular machine and pathname by which a file can be accessed, and transmitted from server to end user using HTTP. Codes, called tags, embedded in an HTML document associate particular words and images in the

25 document with URLs so that a user can access another file, which may literally be halfway around the world, at the press of a key or the click of a mouse. These files may contain text (in a variety of fonts and styles), graphics images, movie files, media clips, and sounds as well as Java applets, ActiveX controls, or other embedded software programs that execute when the user activates them. A user visiting a Web page also

may be able to download files from an FTP site and send messages to other users via email by using links on the Web page.

A server providing a WWW site, as the server described in more detail in conjunction with FIGURE 2 may, is a computer connected to the Internet having

5 storage facilities for storing hypertext documents for a WWW site and running administrative software for handling requests for the stored hypertext documents. A hypertext document normally includes a number of hyperlinks, i.e., highlighted portions of text which link the document to another hypertext document possibly stored at a WWW site elsewhere on the Internet. Each hyperlink is associated with a URL that

10 provides the location of the linked document on a server connected to the Internet and describes the document. Thus, whenever a hypertext document is retrieved from any WWW server, the document is considered to be retrieved from the WWW. As is known to those skilled in the art, a WWW server may also include facilities for storing and transmitting application programs, such as application programs written in the

15 JAVA programming language from Sun Microsystems, for execution on a remote computer. Likewise, a WWW server may also include facilities for executing scripts and other application programs on the WWW server itself.

A user may retrieve hypertext documents from the WWW via a WWW browser application program located on a wired or wireless device. A WWW browser,

20 such as Netscape's NAVIGATOR® or Microsoft's INTERNET EXPLORER®, is a software application program for providing a graphical user interface to the WWW. Upon request from the user via the WWW browser, the WWW browser accesses and retrieves the desired hypertext document from the appropriate WWW server using the URL for the document and HTTP. HTTP is a higher-level protocol than TCP/IP and is

25 designed specifically for the requirements of the WWW. HTTP is used to carry requests from a browser to a Web server and to transport pages from Web servers back to the requesting browser or client. The WWW browser may also retrieve application programs from the WWW server, such as JAVA applets, for execution on a client computer.

FIGURE 2 shows an exemplary server that may operate to provide a WWW site, among other things. When providing a WWW site, server 200 transmits WWW pages to the WWW browser application program executing on requesting devices to carry out this process. For instance, server 200 may transmit pages and 5 forms for receiving information about a user, such as address, telephone number, billing information, credit card number, etc. Moreover, server 200 may transmit WWW pages to a requesting device that allow a consumer to participate in a WWW site. The transactions may take place over the Internet, WAN/LAN 100, or some other communications network known to those skilled in the art.

10                  Those of ordinary skill in the art will appreciate that the server 200 may include many more components than those shown in FIGURE 2. However, the components shown are sufficient to disclose an illustrative environment for practicing the present invention. As shown in FIGURE 2, server 200 is connected to WAN/LAN 100, or other communications network, via network interface unit 210. Those of 15 ordinary skill in the art will appreciate that network interface unit 210 includes the necessary circuitry for connecting server 200 to WAN/LAN 100, and is constructed for use with various communication protocols including the TCP/IP protocol. Typically, network interface unit 210 is a card contained within server 200.

                Server 200 also includes processing unit 212, video display adapter 214, 20 and a mass memory, all connected via bus 222. The mass memory generally includes random access memory ("RAM") 216, read-only memory ("ROM") 232, and one or more permanent mass storage devices, such as hard disk drive 228, a tape drive (not shown), optical drive 226, such as a CD-ROM/DVD-ROM drive, and/or a floppy disk drive (not shown). The mass memory stores operating system 220 for controlling the 25 operation of server 200. It will be appreciated that this component may comprise a general purpose server operating system as is known to those of ordinary skill in the art, such as UNIX, LINUX™, or Microsoft WINDOWS NT®. Basic input/output system ("BIOS") 218 is also provided for controlling the low-level operation of server 200.

                The mass memory as described above illustrates another type of 30 computer-readable media, namely computer storage media. Computer storage media

may include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information, such as computer readable instructions, data structures, program modules or other data. Examples of computer storage media include RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a computing device.

5 memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by a computing device.

The mass memory may also store program code and data for providing a  
10 WWW site. More specifically, the mass memory may store applications including  
WWW server application program 230, and programs 234. WWW server application  
program 230 includes computer executable instructions which, when executed by server  
200, generate WWW browser displays, including performing the logic described above.  
Server 200 may include a JAVA virtual machine, an SMTP handler application for  
15 transmitting and receiving email, an HTTP handler application for receiving and  
handing HTTP requests, JAVA applets for transmission to a WWW browser executing  
on a client computer, and an HTTPS handler application for handling secure  
connections. The HTTPS handler application may be used for communication with an  
external security application to send and receive sensitive information, such as credit  
20 card information, in a secure fashion.

Server 200 also comprises input/output interface 224 for communicating  
with external devices, such as a mouse, keyboard, scanner, or other input devices not  
shown in FIGURE 2. Likewise, server 200 may further comprise additional mass  
storage facilities such as optical drive 226 and hard disk drive 228. Hard disk drive 228  
25 is utilized by server 200 to store, among other things, application programs, databases,  
and program data used by WWW server application program 230. For example,  
customer databases, product databases, image databases, and relational databases may  
be stored.

FIGURE 3 depicts several components of client computer 300. Those of  
30 ordinary skill in the art will appreciate that client computer 300 may include many more

components than those shown in FIGURE 3. However, it is not necessary that those generally-conventional components be shown in order to disclose an illustrative embodiment for practicing the present invention. As shown in FIGURE 3, client computer 300 includes network interface unit 302 for connecting to a LAN or WAN, or 5 for connecting remotely to a LAN or WAN. Those of ordinary skill in the art will appreciate that network interface unit 302 includes the necessary circuitry for such a connection, and is also constructed for use with various communication protocols including the TCP/IP protocol, the particular network configuration of the LAN or WAN it is connecting to, and a particular type of coupling medium. Network interface 10 unit 302 may also be capable of connecting to the Internet through a point to point protocol ("PPP") connection or a serial line internet protocol ("SLIP") connection as known to those skilled in the art.

Client computer 300 also includes BIOS 326, processing unit 306, video display adapter 308, and memory. The memory generally includes RAM 310, ROM 15 304 and a permanent mass storage device, such as a disk drive. The memory stores operating system 312 and programs 334 for controlling the operation of client computer 300. The memory also includes WWW browser 314, such as Netscape's NAVIGATOR® or Microsoft's INTERNET EXPLORER® browsers, for accessing the WWW. It will be appreciated that these components may be stored on a 20 computer-readable medium and loaded into memory of client computer 300 using a drive mechanism associated with the computer-readable medium, such as a floppy disk drive (not shown), optical drive 316, such as a CD-ROM/DVD-ROM drive, and/or hard disk drive 318. Input/output interface 320 may also be provided for receiving input from a mouse, keyboard, or other input device. The memory, network interface 25 unit 302, video display adapter 308, and input/output interface 320 are all connected to processing unit 306 via bus 322. Other peripherals may also be connected to processing unit 306 in a similar manner.

As will be recognized from the discussion below, aspects of the invention may be embodied on routers 110, on server 200, on client computer 300, or 30 on some combination of the above. For example, programming steps may be contained

in programs 334 and/or programs 234. Caching may be performed by one or more servers, such as server 200.

Illustrative Caching System

5 FIGURE 4 illustrates an exemplary environment in which the invention operates in which a virtual server array is connected to a server over a communications medium, according to one embodiment of the invention. The environment includes clients 410-413, WAN/LAN 100, cache 500, server 425, and virtual server array 430.

10 Clients 410-413 are any devices capable of connecting with WAN/LAN 100. Such devices may include devices that typically connect using a wired communications medium such as personal computers, multiprocessor systems, microprocessor-based or programmable consumer electronics, network PCs, and the like. Such devices may also include devices that typically connect using a wireless communications medium such as cell phones, smart phones, pagers, walkie talkies, 15 radio frequency (RF) devices, infrared (IR) devices, CBs, integrated devices combining one or more of the preceding devices, and the like. In addition, clients 410-413 may also include devices that are capable of connecting using a wired or wireless communication medium such as PDAs, POCKET PCs, wearable computers, and other devices mentioned above that are equipped to use a wired and/or wireless 20 communications medium. An exemplary client that may connect with WAN/LAN 100 is client computer 300 of FIGURE 3.

25 Server 425 receives, among other things, requests for content. Content includes information that may be found on one or more WWW servers such as Web pages, documents, files, databases, FTP data, or other information. After receiving a request for content, server 425 may forward the request to a cache in cache 500 or to a server in virtual server array 430. Server 425 may determine where to forward the request to using many factors including where the request originates from, what the request is requesting, load on servers in virtual server array 430, and/or other factors. Requests for content may come from a device coupled to WAN/LAN 100, a server in 30 virtual server array 430, a cache in cache 500, or some other requestor.

Cache 500 and its interaction with server 425 are described in more detail in conjunction with FIGURE 5. Briefly, cache 500 receives requests for content from server 425. If cache 500 does not have a copy of the content requested, it may request the content from server 425 which may then forward the request to a server in 5 virtual server array 430. Server 425 keeps track of which content is being frequently requested which is stored in cache 500 to speed retrieval of the content.

Virtual server array 430 may include one or more WWW servers, such as server 200 of FIGURE 2. These servers may be physically connected to server 425 as shown in FIGURE 4 or they may be connected somewhere on WAN/LAN 100 as a 10 remote computer as shown in FIGURE 1. Generally, server 425 causes virtual server array 430 to appear as one WWW server. That is, a client, such as client 410, requesting content from server 425 for a Web site will typically be unaware that the request is serviced by a server in virtual server array 430. Server 425 may translate 15 requests from the client and send them to a particular server and then translate responses from the server to make it appear that server 425 itself satisfied the request.

Virtual server array 430 may serve content for more than one vendor. For example, a group of vendors may each run their Web sites using one or more servers in virtual server array 430.

FIGURE 5 shows a caching mechanism for speeding access to content, 20 according to one embodiment of the invention. The mechanism includes server 425, hash table 505, and cache 500. Cache 500 includes regular caches 510-512 and hot caches 515-516.

Server 425 may be connected to the caches in cache 500 through one or 25 more communications mediums. Server 425 may include hash table 505 or may be connected to another server or device including hash table 505.

Hash table 505 is described in more detail in conjunction with FIGURE 6. Briefly, hash table 505 may be utilized to track requests to determine the frequency with which each content is requested. Server 425 may use hash table 505 to determine to which cache in cache 500 it should forward a request for content.

As described in conjunction with FIGURE 4, when server 425 receives a request for content, it determines whether to forward the request to a cache in cache 500 or to a server in virtual server array 430. Server 425 may use several factors in making its determination. One factor server 425 may use is information about the requestor, 5 such as where the request comes from. When a request comes from a regular cache, such as regular cache 510, this generally indicates that the cache does not contain information that was previously requested. When a cache does not contain information requested, this is known as a cache miss. When server 425 receives a request for content from a regular cache, it may forward this request to a server in virtual server 10 array 430. When the server returns the content, the requesting cache can then cache the information and send it to a device that requested it. The next time the cache receives a request for the same content, provided it has not eliminated a copy of the content, it may send its copy of the information to the requestor. A cache might eliminate a copy of content if the content has not been requested for a period of time or if storage space is 15 needed for other content.

A requestor may be a hot cache. When server 425 receives a request for content from a hot cache, such as hot cache 515, it may forward the request to a cache, such as one of regular caches 510-512. Generally, the regular cache contains the content and can quickly send the hot cache a copy of it. This will be understood by a 20 brief description of how server 425 distributes requests to caches in cache 500. When server 425 receives a request for a content from an outside source, e.g. one of clients 410-413, it looks in hash table 505 to determine if the content has been recently frequently requested (also known as hot).

Content may be hot, for example, if the frequency of requests for it 25 during a time period exceed a certain threshold. Content may also be considered hot, for example, during a period of cool down. For example, if a large number of requests for content come in a short period followed by a brief lull in activity followed by another large number of requests, although “cool” during the lull, the content may not be marked as cool until a selected amount of time has passed. Some content may be

specified to be hot regardless of the frequency of requests to it. For example, a vendor may wish to mark some of its Web pages as always hot.

When server 425 determines that the request is not for hot content, it hashes the URL of the request, e.g. <http://www.company1.com/support>, to determine a regular cache to which the request should be forwarded. A hashing function might take the ASCII (American Standard Code for Information Interchange) value of each letter in the URL, and perform some mathematical operations to obtain a hash value. This hash value could then be divided by the number of available regular caches to obtain a remainder. The remainder could then be used as the cache to which to send the request.

10 Hashing in this fashion may cause requests for cool content to be hashed to the same value as requests for hot content. Server 425 may use the table shown in FIGURE 6 in addition to the hashing function to send a request for hot content to a hot cache and a request for cool content to a cool cache as described in more detail in conjunction with FIGURE 6.

15 After server 425 has obtained a hash value corresponding to a regular cache, when the content is cool, it sends the request to the regular cache. If server 425 receives several requests for the same URL in a short period of time, it may determine that the content associated with the URL should be marked as hot. While content is marked hot, server 425 generally sends requests for the content to one of hot caches  
20 515-516, such as hot cache 515. Usually, the first time hot cache 515 receives a request for the content, it does not have the content, but the regular cache that was receiving requests for the content just prior to the request being sent to hot cache 515 usually does. Therefore, when a hot cache, such as hot cache 515, requests the content because of a cache miss, server 425 may determine that a regular cache should contain the  
25 content and forward the request to the regular cache. This eliminates sending a request to a server in virtual server array 430 which avoids potential latency, especially when the server is not connected directly to server 425 (as seen in FIGURE 10).

Another factor server 425 may use to forward the request to a server a cache is information about what is requested. For example, a request may include  
30 information that indicates that the request is for dynamically generated content. For

example, the request may be for stock prices. Stock prices often change. Extensions within the request may indicate that a program or script will run to generate the content. For example, a request may include a reference including an extension of .asp, .cfm, .cgi, .jsp, .js, .php, etc. These extensions are commonly used to cause a program or 5 script to run to generate dynamic content. Server 425 may determine that such requests should not be cached and send them to a server in virtual server array 430.

It will be appreciated that server 425 may take into account other factors when determining whether to forward the request to a cache or a server without departing from the spirit or scope of this invention.

10 It will also be appreciated that there may be more than two levels of cache. That is, there may be a hierarchy of caches. Which cache content is forwarded to may depend on the frequency with which the content is requested or other factors as described above. For example, in a three level cache system, highly requested content may be forwarded to a highest level cache, regularly requested content may be 15 forwarded to a medium level cache, and seldom requested content may be forwarded to a low level cache. When a cache at any level does not contain the content, it sends a request to the forwarder which may then forward the request to a lower level cache or content server as appropriate.

Regular caches 500-512 and hot caches 515-516 are connected to server 20 425. They may be implemented as a special purpose electronic device or using a server such as server 200. More than one cache may be on one electronic device or server. For example, regular cache 500 and hot cache 515 may be on one server while regular caches 510-512 and hot cache 516 may be on another server. A cache may be spread over more than one electronic device or server. For example, two or more servers may 25 act logically as one or more caches.

#### Exemplary Data Structure

FIGURE 6 illustrates an exemplary table that may be used to maintain information about content requests, according to one embodiment of the invention. The 30 table includes a content field, a hits field, and a hot field. The content field shown in

the figure gives a URL in ASCII for the content requested. In another embodiment of the invention, the content field contains a numerical value corresponding to a hash value of the URL. The content field may be indexed to allow faster searching.

Although the table shows only one reference to content per row, the  
5 content field may include multiple references to different content per row or a link to a record that contains multiple references to different content. The set of different content referenced by a row is known as a content set. A request for any of the content in a content set may be used to increment the hits field associated with the content set and in hot/cold calculations. A content set may be considered hot even if some of the content  
10 in the content set is not requested. For example, even though some of the content in a content set may not be requested, other content in the content set may be frequently requested. When performing hot/cold calculations, the hits (requests) for all the content in the content set may be summed. This allows a vendor to group content together.

Throughout this document references are made to content and content  
15 set. Where it makes sense, a reference to content should be construed to mean additionally or alternatively content in a content set or a content set associated with the content. For example, a reference to a request for content should also be construed to mean a request for content in a content set. A reference to tracking which content is frequently requested should also be construed as a reference to tracking the frequency  
20 of requests for content included in a content set associated with the content.

The hits field corresponds to the frequency with which the content is requested. It may include the actual number of times the content was requested. It may be reset with each time period. It may be a weighted rate of hits, with more recent hits being weighted more heavily.

25 The hot field gives an indication of how hot the content is. It may be a binary, yes/no, type of field in which a zero indicates cool and a one indicates hot. It may take on more than two values. For example, to implement a “cool” down period, when the frequency of requests for a content surpasses a hot threshold, the hot field may be set to 3. After a period passes, if the frequency of requests during the period are  
30 below a cool threshold, the hot field may be decremented. While the hot field stays

above zero, requests for the content are forwarded to a hot cache, even if in a previous time period the frequency of requests for the content were below the cool threshold.

The table shown in FIGURE 6 may be used with the hashing function previously discussed to send requests for hot content to a hot cache while sending 5 requests for cool content to a regular cache. Suppose that the hashing function hashed the two URLs, [www.company1.com/news](http://www.company1.com/news) and [www.company2.com/support](http://www.company2.com/support) to the same value. By looking in the table, server 425 may determine that requests for [www.company1.com/news](http://www.company1.com/news) should be directed to a hot cache whereas requests for [www.company2.com/support](http://www.company2.com/support) should be directed to a regular cache indicated by the 10 hash value. After server 425 determines that a request should be directed to a hot cache, it determines to which hot cache the request should be forwarded. Server 425 may be configured to use one of several approaches including round-robin, least-loaded, least connections, priority of requestor and speed of cache, fastest response time, or any other load balancing algorithm.

15

#### Illustrative Method for Forwarding Requests for Content

FIGURE 7 shows a flow chart for receiving and forwarding requests for content, according to one embodiment of the invention. The process begins at block 705 after a requestor, such as one of clients 410-413 of FIGURE 4, requests content.

20

At block 710, the request for content is received. For example, referring to FIGURE 4, server 425 receives a request for a Web page that is located on one of the servers in virtual server array 430.

25

At block 715, a determination is made as to whether the content is dynamic. For example, the request may be for content that ends in .asp, .cfm, .cgi, .jsp, .js, .php, or other content that server 425 knows is dynamic. If the request is for dynamic content processing branches to block 720; otherwise, processing branches to block 725.

At block 720, the request is forwarded to a content server without being forwarded to a cache. This avoids caching information that may change for each

requestor. For example, referring to FIGURE 4, server 425 forwards the request to a server in virtual server array 430.

At block 725, the request is hashed to find a hash value corresponding to content. This hash value may be used later to update the frequency with which a content is requested when the request does not come from a cache. For example, referring to FIGURE 4, server 425 takes the request for <http://www.company1.com/support> and hashes it into an index into the table shown in FIGURE 6.

At block 730, a determination is made as to whether the request comes from a cache. When it does, processing branches to block 740; otherwise, processing branches to block 735. This may occur to avoid the double counting of requests. For example, if a site requested content, usually, the hits in the table of FIGURE 6 are increased. If, however, a cache did not contain the content, it would request the content from the server. If the server did not check to see if the request comes from a cache, it might increase the hit count again for that content. For example, referring to FIGURE 4, server 425 determines that the request is coming from one of clients 410-413.

At block 735, a hot/cold calculation occurs as described in more detail in conjunction with FIGURE 9. Briefly, a rate of requests is calculated and the content requested may be marked hot or cold depending on the rate. For example, referring to FIGURE 4, server 425 updates the table shown in FIGURE 6 to indicate whether the content requested is hot or cold. In another embodiment of the invention, a process runs periodically to update the hot/cold status of each content in the table of FIGURE 6. In this embodiment, the hot/cold calculation that occurs in response to a request in block 735 may be skipped as it is done by the periodic process.

At block 740, the request is forwarded appropriately as described in more detail in conjunction with FIGURE 8. Briefly, if the request is for hot content, it is forwarded to a hot cache. If the request is for cold content, it is forwarded to a regular cache. If the request is from a regular cache that the request would have been hashed to if cold, the request is forwarded to a content server.

At block 745, the process ends. At this point a request for content has been received. When the request is determined to request dynamic content, it has been forwarded to a content server. Otherwise, a hashing function has been utilized to find an appropriate cache to which the request might be forwarded. If the request is not 5 from a cache, statistics are updated and a hot/cold calculation may be performed. If the request was from a cache, it has been forwarded to another cache or to a content server.

FIGURE 8 illustrates a flow chart for determining where to forward a request for non-dynamic (also called static) content, according to one embodiment of the invention. The process begins at block 805 after a request from a cache has been 10 received.

At block 810, a determination is made as to whether the requestor is the cache that a request for the content would have been hashed to if cold. A cache may incur a cache miss if it does not contain content that has been requested from it. To obtain the content, the cache requests it from a server that forwarded the request to the 15 cache. The server recognizes that a request for content coming from a cache to which the server previously requested the content indicates that the cache does not have the content. If the cache is the cache that a request for the content would have hashed to, processing branches to block 820; otherwise, processing branches to block 815.

Referring to FIGURE 5, server 425 receives a request for content associated with 20 <http://www.company1.com/support> and forwards the request to regular cache 510. Regular cache 510 does not contain the content, so it requests it from server 425. Server 425 recognizes that the cache to which it hashed and forwarded the request does not contain the content, and prepares to forward the request from the cache to a server in virtual server array 430 of FIGURE 4.

25 At block 820, the content is forwarded to a content server. For example, referring to FIGURE 5, server 425 forwards the request to a server in virtual server array 430.

At block 815, a determination is made as to whether the content is hot. If it is, processing branches to block 825; otherwise processing branches to block 830.

For example, referring to FIGURES 5 and 6, server 425 determines that the request for the content associated with http://www.company1.com/support is cold.

At block 825, the request is forwarded to a hot cache. For example, referring to FIGURES 5 and 6, server 425 uses a round-robin approach to forward the 5 request for http://www.company2.com/home to hot cache 515.

At block 830, the request is forwarded to the cache to which it was hashed. For example, referring to FIGURE 5, server 425 forwards the requests for http://www.company1.com/support to regular cache 510.

At block 835, the process returns to the calling process. At this point, 10 the request has been forwarded to a content server, a hot cache, or a regular cache.

FIGURE 9 shows a flow chart for performing a hot/cold calculation, according to one embodiment of the invention. The process begins at block 905 after a request from a requestor other than a cache has been received.

At block 910, the content set indicated by the hash value is located. For 15 example, referring to FIGURE 6, a content set associated with the content requested is located. For purposes of this discussion, assume that each content set shown in the content field of FIGURE 6 includes content associated with children URLs. For example, the content set www.company1.com/home would include www.company1.com/home in addition to www.company1.com/home/FAQ, 20 www.company1.com/home/services, etc. Then, the content set located for a request for www.company1.com/home/services would be www.company1.com/home.

At block 915, the hit count of the content set is incremented. For example, referring to FIGURE 6, the hits field for www.company1.com/home is incremented.

25 At block 920, a determination is made as to whether a selected time period has elapsed since the last hot/cold calculation. When the period has not elapsed, the hot/cold calculation is not performed and processing branches to block 955. Otherwise, processing branches to block 925 to determine whether the content set needs to be marked hot or cold.

At block 925, a determination is made as to whether the content set is marked hot. For example, referring to FIGURE 6, the content set associated with www.company1.com/home is currently marked as cold. When the content set is marked hot, processing branches to block 930; otherwise, processing branches to block 5 935.

At block 930, a determination is made as to whether the hit rate of the content set is less than or equal to a cool threshold. If it is, processing branches to block 940; otherwise, processing branches to block 955. The requests for content in a content set may fall below a cool threshold indicating that requests for the content should no 10 longer be serviced by hot caches. This block determines whether the rate of requests has fallen below the threshold. Note, that a different hot and cool threshold may be set for each content set or a hot and cool threshold may be set globally to apply to all content sets.

At block 935, a determination is made as to whether the hit rate of the content is greater than or equal to a hot threshold. When it is, processing branches to block 945; otherwise, processing branches to block 950. It is at this block that a content set that is not currently marked as hot is checked to determine whether it should be marked hot.

At block 940, the content set is marked as cold. For example, referring 20 to FIGURE 6, the hot field for the content set is modified to indicate cold status. For example, the hit frequency for www.company1.com/news may fall from 15 to 5, dropping below the cold threshold of 6 per period, and the hot field is changed from 1 to 0 to indicate that the content is cold.

At block 945, the content set is marked as hot. For example, referring to 25 FIGURE 6, the hot field for the content set is modified to indicate hot status. For example, the hit frequency for www.company1.com/products may increase by 3 to 10, exceed the hot threshold of 9 per period, and the hot field is changed from 0 to 1 to indicate that the content is hot.

At block 950, the process returns to the calling process. At this point, a 30 content set has been located and the hit count of the content set increased. If a selected

time period has elapsed, the content set is checked to determine whether it needs to be marked hot or cold depending on the number of requests for the past period and a hot and cold threshold. Using the now-current hot field, the calling process may now forward the request to a hot or regular cache appropriately.

5

#### Another Exemplary Configuration of a Caching System

FIGURE 10 illustrates an exemplary environment in which the invention operates in which a virtual server array is connected to a server over a network, according to one embodiment of the invention. A difference between the environment 10 shown in FIGURE 4 and the environment shown in FIGURE 10 is that server 425 reaches virtual server array 430 over WAN/LAN 100 rather than being more directly connected with it. Virtual server array 430 may include many physical servers which may be connected to WAN/LAN 100 at various places. That is, even the servers within virtual server array 430 do not need to be physically located together.

15 In light of this disclosure, it will be recognized that many variations of the above process may be implemented without departing from the spirit or scope of this invention.

20 The various embodiments of the invention may be implemented as a sequence of computer implemented steps or program modules running on a computing system and/or as interconnected machine logic circuits or circuit modules within the computing system. The implementation is a matter of choice dependent on the performance requirements of the computing system implementing the invention. In 25 light of this disclosure, it will be recognized by one skilled in the art that the functions and operation of the various embodiments disclosed may be implemented in software, in firmware, in special purpose digital logic, or any combination thereof without deviating from the spirit or scope of the present invention.

30 The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims hereinafter appended.